REMARKS/ARGUMENTS

This paper is being provided in response to the October 6, 2006 Office Action for the above-referenced application. In this response, Applicant has amended Claims 2, 15, 25, 38, 44 and 45 in order to clarify that which Applicant deems to be the claimed invention. Applicant respectfully submits that the amendments to the claims are all supported by the originally filed application.

In the Office Action it is noted that Claim 43 was missing. Applicant respectfully submits that Claims 44-46 were misnumbered in the originally filed application and should have been respectively numbered as Claim 43-45. Accordingly, Applicant has renumbered the originally filed Claims 44-46 as Claims 43-45, respectively, herein.

Applicant thanks the Examiner regarding the indication of allowability regarding the subject matter of Claims 21-23 and 44-46 (now renumbered as 43-45). Applicant also thanks the Examiner for the indication of the allowability regarding the subject matter of dependent Claims 15-16 and 38-39 if rewritten in independent form including all limitations of the base claim and any intervening claims. Accordingly, Applicant has rewritten Claims 15 and 38 in accordance with remarks set forth in the Office Action. Applicant respectfully submits that Claims 15 and 38, and Claims 16 and 39 that depend, respectively, therefrom, are also now in condition for allowance.

In response the rejection of Claim 2 under 35 U.S.C. 112, second paragraph, as being indefinite, Applicant has amended Claim 2 in accordance with remarks set forth in the Office

Action. Accordingly, Applicant respectfully requests that the rejection be reconsidered and withdrawn.

The rejection of Claims 1-3, 12, 24-26 and 35 under 35 U.S.C. 102(b) as being anticipated by Whiteside et al., (U.S. Patent No. 4,330,826, hereinafter referred to as "Whiteside") is hereby traversed and reconsideration thereof is respectfully requested.

Claim 1 recites a method for synchronizing a plurality of endpoints in a data storage system at a first synchronization point, the method comprising: performing, by a first endpoint, a synchronization start operation wherein a first message is sent from said first endpoint to one or more other endpoints in the data storage system, said first message including a first key value corresponding to said first synchronization point representing a current processing state of said first endpoint; determining, by said first endpoint, a timeout period; determining, by said first endpoint using processing state information as reported to said first endpoint by other endpoints, whether synchronization with a selected portion of said one or more other endpoints at said first synchronization point has been accomplished within said timeout period; and if said first endpoint determines that synchronization has not been accomplished within said timeout period, sending a second message to said one or more other endpoints indicating that said first endpoint is at another synchronization point different from said first synchronization point. Claims 2, 3, and 12 depend from Claim 1.

Claim 24 recites a computer program product for synchronizing a plurality of endpoints in a data storage system at a first synchronization point, the computer program product comprising: executable code that performs, by a first endpoint, a synchronization start operation wherein a first message is sent from said first endpoint to one or more other endpoints in the data storage system, said first message including a first key value corresponding to said first synchronization point representing a current processing state of said first endpoint; executable code that determines, by said first endpoint, a timeout period; executable code that determines, by said first endpoint using processing state information as reported to said first endpoint by other endpoints, whether synchronization with a selected portion of said one or more other endpoints at said first synchronization point has been accomplished within said timeout period; and executable code that, if said first endpoint determines that synchronization has not been accomplished within said timeout period, sends a second message to said one or more other endpoints indicating that said first endpoint is at another synchronization point different from said first synchronization point. Claims 25, 26, and 35 depend from Claim 24.

Whiteside discloses a synchronizer for synchronizing operation of computers. A synchronizer is associated with each computer and regularly generates signals to initiate the operations in its own computer. (See Abstract). Whiteside's Figure 17 is a functional block diagram of a synchronizer. The Check Sampling Timer Module 464 regularly checks the remaining time in the sampler period timer in order to detect the end of the sample period. At the end of each sampling period, it restarts the sampling period timer for the next sampling period, sends a sampling number message to Transmitter 212, and checks if a voted sampling number was obtained during the sampling period. (Col. 47, Lines 16-43; Figure 17). The find sampling number agreement module 466 generates a voted sampling number when sampling number messages received from like synchronizers in a predetermined number of computers contain the same sampling number. When a voted sampling number is obtained, the synchronizer generates a signal initiating in its own computer the operations which are to be synchronized among the computers. When a voted sampling number is obtained, the module 466 adjusts the remaining time in the sampling period timer so that its sampling period coincides with the sampling period of the other computers and records the "voting sampling number" as its own current sampling number. (See Abstract; Figure 17; Col. 2, Lines 28-41; Col. 47, Lines 44-59). If a voted sampling number is not obtained, the check sampling module generates a restart signal causing the synchronizer module to reinitialize the synchronizer. (Abstract; Col. 2, Lines 46-52; Col. 47, Lines 39-42).

Applicant's Claim 1 is neither disclosed nor suggested by Whiteside in that Whiteside neither discloses nor suggests at least the features of ***a method for synchronizing a plurality of endpoints in a data storage system at a first synchronization point, the method comprising: performing, by a first endpoint, a synchronization start operation wherein a first message is sent from said first endpoint to one or more other endpoints in the data storage system, said first message including a first key value corresponding to said first synchronization point representing a current processing state of said first endpoint; ... determining, by said first endpoint using processing state information as reported to said first endpoint by other endpoints, whether synchronization with a selected portion of said one or more other endpoints at said first synchronization point has been accomplished within said timeout period; and if said first endpoint determines that synchronization has not been accomplished within said timeout period, sending a second message to said one or more other endpoints indicating that said first endpoint is at another synchronization point different from said first synchronization point, ...*** as set forth in Claim 1.

As described above, Whiteside discloses a synchronizer which, at the end of each sampling period, restarts the sampling period timer for the next sampling period, sends a sampling number message to Transmitter 212, and checks if a "voted sampling number" was received during the sampling period from like synchronizers in a predetermined number of computers. The "voted sampling number" is the sampling number included in messages received from a predetermined number of computers. When a voted sampling number is obtained indicating that a predetermined number of received messages contain a same sampling number, the synchronizer generates a signal initiating in its own computer the operations which are to be synchronized among the computers. When no voted sampling number is obtained, the synchronizer is restarted.

The Office Action at pages 4-5 suggests that making a determination as to whether "synchronization", as recited in Claim 1, has been accomplished is analogous to checking if other computers contain the same sampling number message as the first computer. The Office Action also suggests that Whiteside performs the foregoing synchronization (e.g., that Whiteside checks to see if the other computers contain the same sampling number message as the first computer), and that Whiteside performs conditional processing based on the foregoing. Furthermore, the Office Action indicates that, upon determining that synchronization has not been accomplished (e.g., if a determination is made that the other computers do not contain the same sampling number message as the first computer), Whiteside restarts the synchronizer. Applicant respectfully submits that Whiteside neither discloses nor suggests performing the foregoing synchronization determination and conditionally performing processing based on the determination.

Whiteside discloses determining whether a predetermined number of received messages have a same sampling number (e.g., voted sampling number). Whiteside's synchronization is determined based on whether the sampling number contained in a predetermined number of received messages is the same, not whether the sampling number contained in a predetermined number of received messages is the same as a sampling number of the receiving synchronizer. Whiteside discloses performing conditional processing based on a determination of the foregoing, not based on any comparison between a sampling number of a receiving synchronizer and a same sampling number of other computers. The "second message" (e.g., restart signal) of Whiteside is sent if a voting sample number is not obtained during the preceding period. Whether the restart signal is sent is not based on a comparison of the sampling number of the receiving synchronizer with respect to the voting sampling number. Rather, the restart signal is sent if a predetermined number of received messages did not have the same sampling number (e.g., no voting sampling number is obtained). In other words, "synchronization" in Whiteside is achieved if a predetermined number of received messages contain the same sampling number (e.g., a voting sampling number is obtained) and even when that same sampling number does not match the sampling number of the synchronizer processing the received messages. In contrast to Whiteside, Applicant's Claim 1 determines ***whether synchronization with a selected portion of said one or more other endpoints at said first synchronization point has been accomplished within said timeout period*** in which the "first synchronization point" is the synchronization point of the first performing step representing a current processing state of the first endpoint. Whiteside performs no such determination with respect to a processing state of the first end point and one or more other endpoints. Rather, Whiteside performs a determination as to whether a predetermined number of computers sending messages to a first computer are synchronized with each other (e.g., have a same sampling number) independent of whether this same sampling number matches a sampling number of the "first computer" (e.g., independent of whether the predetermined number of computers have a synchronized sampling number with respect to the first computer receiving the messages).

As such, Whiteside does not disclose or fairly suggest ***performing, by a first endpoint, a synchronization start operation wherein a first message is sent from said first endpoint to one or more other endpoints in the data storage system, said first message including a first key value corresponding to said first synchronization point representing a current processing state of said first endpoint; ... determining, by said first endpoint using processing state information as reported to said first endpoint by other endpoints, whether synchronization with a selected portion of said one or more other endpoints at said first synchronization point has been accomplished within said timeout period; and if said first endpoint determines that synchronization has not been accomplished within said timeout period, sending a second message to said one or more other endpoints indicating that said first endpoint is at another synchronization point different from said first synchronization point***, as set forth in Claim 1.

For at least the foregoing reasons, Applicant's Claim 1, and claims that depend therefrom, are neither disclosed nor suggested by Whiteside.

Claim 24 recites features similar to those pointed out above which are included in Claim 1 and are neither disclosed nor suggested by Whiteside. Thus, Claim 24, and claims that depend therefrom, are also neither disclosed nor suggested by Whiteside for reasons similar to those set forth regarding Claim 1.

In view of the foregoing, Applicant respectfully requests that the rejection be reconsidered and withdrawn.

The rejection of Claims 4-9, 13-14, 27-32 and 36-37 under 35 U.S.C. 103(a) as being unpatentable over Whiteside in view of Laudon, (U.S. Patent No. 5,680,576, hereinafter referred to as "Laudon") is hereby traversed and reconsideration thereof is respectfully requested.

Claims 4-9 and 13-14 depend from Claim 1. Claims 27-32 and 36-37 depend from Claim 24. For reasons set forth above, Claims 1 and 24, and claims that depend therefrom, are patentable over Whiteside. For reasons set forth below, Applicant respectfully submits that combining Whiteside with Laudon also neither discloses nor suggests Claims 1 and 24, and claims that depend therefrom.

Laudon relates to cache coherency and more particularly, to a directory-based coherence protocol allowing efficient dropping of clean-exclusive (CEX) data. (Col. 1, Lines 21-23). When a CEX object is replaced, it does not need to be written back to main storage since the copy in the main storage is the same. (Col. 1, Line 59-Col. 2, Line 1). Laudon's background discloses that keeping copies of cached information is important. Many protocols have been devised where coherence is maintained by broadcasting all operations to all caches. (Col. 1, Lines 26-40). To solve scalability problems, cache coherence based on directories is employed. The directory keeps information for each object. Each time a requestor wishes to access the object, the requestor interrogates the directory which determines the operations necessary to maintain coherence on the object. (Col. 1, Lines 41-50).

Features of Applicant's Claim 1 which are neither disclosed nor suggested by Whiteside are set forth above. Laudon also appears silent regarding the foregoing recited features of Claim 1. Thus, combining Whiteside with Laudon does not overcome the deficiencies of Whiteside with respect to Applicant's Claim 1.

Applicant's Claim 24 recites features similar to those set forth above regarding Claim 1. Thus, for reasons similar to those set forth above regarding Claim 1, Applicant's Claim 24 is also neither disclosed nor suggested by the references.

In view of the foregoing, Applicant respectfully requests that the rejection be reconsidered and withdrawn.

The rejection of Claims 10-11 and 33-34 under 35 U.S.C. 103(a) as being unpatentable over Whiteside in view of Oberlin et al., (U.S. Patent No. 5,434,995, hereinafter referred to as "Oberlin") is hereby traversed and reconsideration thereof is respectfully requested.

Claim 10-11 depend from Claim 1. Claims 33-34 depend from Claim 24. For reasons set forth above, Claims 1 and 24, and claims that depend therefrom, are neither disclosed nor suggested by Whiteside. For reasons set forth below, combining Whiteside with Oberlin also does not disclose or suggest Claims 1 and 24, and claims that depend therefrom.

Page 10 of the Office Action cites Oberlin as support for disclosing a synchronization mask containing a number of bits equal to the number of processing elements to enable or disable the synchronization circuit. (Figure 7; Col. 9, Line 45+).

Features of Applicant's Claim 1 which are neither disclosed nor suggested by Whiteside are set forth above. Oberlin also appears silent regarding the foregoing recited features of Claim 1. Thus, combining Whiteside with Oberlin does not overcome the deficiencies of Whiteside with respect to Applicant's Claim 1.

Applicant's Claim 24 recites features similar to those set forth above regarding Claim 1. Thus, for reasons similar to those set forth above regarding Claim 1, Applicant's Claim 24 is also neither disclosed nor suggested by the references.

In view of the foregoing, Applicant respectfully requests that the rejection be reconsidered and withdrawn.

The rejection of Claims 17-20 and 40-42 under 35 U.S.C. 103(a) as being unpatentable over Whiteside in view of Elkington et al., (U.S. Patent No. 6,571,324, hereinafter referred to as "Elkington") is hereby traversed and reconsideration thereof is respectfully requested.

Claim 17-20 depend from Claim 1. Claims 40-42 depend from Claim 24. For reasons set forth above, Claims 1 and 24, and claims that depend therefrom, are neither disclosed nor suggested by Whiteside. For reasons set forth below, combining Whiteside with Elkington also does not disclose or suggest Claims 1 and 24, and claims that depend therefrom.

Page 11 of the Office Action cites Elkington as support for disclosing a first synchronization point (i.e., the warmswap process) in which the mirror write is disabled by not writing to the failed cache and only writing to the good cache, and each cache is on a different memory board. (See Abstract and Figure 1). The Office Action also states that Elkington teaches that the first synchronization point (i.e., the warmswap process) represents an event corresponding to one of: enabling and disabling operations to a hardware element by disabling operations of the hardware element, the bad cache, by bringing it off-line, and enabling operations of the hardware element, the new replacement cache, by bringing the new cache online. (See Abtract).

Features of Applicant's Claim 1 which are neither disclosed nor suggested by Whiteside are set forth above. Elkington also appears silent regarding the foregoing recited features of Claim 1. Thus, combining Whiteside with Elkington does not overcome the deficiencies of Whiteside with respect to Applicant's Claim 1.

Applicant's Claim 24 recites features similar to those set forth above regarding Claim 1. Thus, for reasons similar to those set forth above regarding Claim 1, Applicant's Claim 24 is also neither disclosed nor suggested by the references.

In view of the foregoing, Applicant respectfully requests that the rejection be reconsidered and withdrawn.

Based on the above, Applicant respectfully requests that the Examiner reconsider and withdraw all outstanding rejections and objections. Favorable consideration and allowance are earnestly solicited. Should there be any questions after reviewing this paper, the Examiner is invited to contact the undersigned at 508-898-8604.

Respectfully submitted,
MUIRHEAD AND SATURNELLI, LLC

Anne E. Saturnelli
Registration No. 41,290

MUIRHEAD AND SATURNELLI, LLC
200 Friberg Parkway, Suite 1001
Westborough, MA 01581
Tel: (508)898-8604
Fax: (508)898-8602

Date: December 28, 2006